algorithms

MDPI

Article

Exact Pareto Dynamic Programming and Deep Fitted Q-Iteration for Cost-Constrained Reliability Design of Directed Polyomino-Chain Networks

Mushtaq K. Abdalrahem [1,†], Ahmad H. Alkasasbeh [2,*,†], Laila Azami [3,†] and Nader Jafari Rad [3,†]

[1] College of Pharmacy, University of Al-Ameed, Iraq
[2] Oryx University in Partnership with Liverpool John Moores University, Doha P.O. Box 12253, Qatar
[3] Department of Mathematics, Shahed University, Tehran 3319118651, Iran
* Correspondence: ahmad.alkasasbeh@oryx.edu.qa
† These authors contributed equally to this work.

Abstract

We study cost-constrained two-terminal reliability design in directed polyomino-chain networks with linear and kink attachments. Using exact three-dimensional reliability dynamics, continuation words are represented by alpha-vectors, yielding a dominance-pruned dynamic program and a heterogeneous cost–alpha Pareto frontier for hard-budget and penalized optimization. The model has positive homogeneous value functions and scale-invariant optimal policies. For position-dependent kink costs, a cross-instance Markov state generally requires the ordered suffix of remaining costs. Motivated by these results, we develop finite-horizon Deep Fitted Q-Iteration (Deep FQI) using normalized states, exact model-based Bellman targets, backward training, and an action-gap-aware loss. Exact optimizers provide independent evaluation, not training labels. On 160 unseen heterogeneous-cost test instances, exact Pareto optimization agreed with brute-force enumeration to within 6.7×10^{-16}. Across five training seeds, Deep FQI reduced mean absolute optimality gaps relative to a myopic feasible policy by 31.8%, 48.0%, and 72.2% on the principal, high-r, and extrapolation test families, respectively; reductions relative to a random-feasible baseline were 49.5%, 70.7%, and 81.2%. Exact-design recovery remained modest. The learned policy is therefore an approximate amortized decision rule, complementary to exact optimization; its performance beyond eight decisions has not been established.

Keywords: state-space reduction; exact optimality certification; heterogeneous attachment costs; action-gap approximation; reproducible computational experiments

MSC: 05C40; 68T05; 68T07; 94C15

Check for updates

Academic Editor: Adele Anna Rescigno

Received: 20 October 2026
Revised: 19 November 2026
Accepted: 23 November 2026
Published: 28 November 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Two-terminal reliability is the probability that two distinguished vertices remain connected after independent component failures. Exact reliability evaluation is computationally difficult on general graphs, but recursively constructed graph families can admit compact state representations that replace explicit enumeration of all operational subgraphs [1,2]. Polyomino chains are a particularly structured example: starting from two adjacent squares, each subsequent square is attached in one of two locally admissible ways, naturally encoded as a binary word. Their recursive geometry has been exploited for topological indices and extremal problems [3–5].

Recent reliability work on directed polyomino-chain networks provides an exact finite-state transfer representation and a lower-dimensional invariant reduction [6,7]. In the present paper, "directed" refers to the prescribed growth orientation of the chain; the two-terminal reliability event itself is ordinary connectivity in the operational underlying network. The reduced dynamics make it possible to ask a different question: rather than evaluate a fixed chain, how should one design the sequence of linear and kink attachments when kink choices have construction costs?

Related work has increasingly used reinforcement learning in reliability, maintenance, network restoration, and resilience planning [8–10]. These studies establish the relevance of sequential learning under uncertainty, but they do not supply an exact cost–alpha dominance certificate for the specific recursively generated attachment decisions considered here. Our research gap is the joint provision of (i) exact reusable cost–reliability frontiers for this graph family and (ii) an amortized policy whose reported errors can be independently certified against the exact optimum. The use of an analytic transition model, rather than a black-box learned reliability surrogate, is another important distinction.

This question combines three issues. First, the design space contains exponentially many attachment words. Second, a locally attractive kink may consume budget that would be more valuable at a later position, so the optimization is genuinely sequential. Third, when costs vary by position, the future cost profile itself affects the optimal decision; a scalar remaining budget is not a sufficient description for a policy shared across different instances.

The paper develops exact and learned methods in a deliberately complementary way. We first use the linear reliability dynamics to derive an alpha-vector representation and an exact dynamic program with componentwise dominance pruning. We then extend the representation to a joint cost–alpha Pareto frontier for heterogeneous costs. Unlike a budget-indexed dynamic program, the resulting frontier can answer all hard-budget queries and all nonnegative soft-cost penalties for a fixed reliability parameter and cost profile after one backward construction.

The exact formulation also reveals a structure useful for learning. The optimal value is positively homogeneous in the reliability state, so the optimal action depends on the normalized state direction rather than its absolute magnitude. Moreover, multiplying all remaining costs and the remaining budget by the same positive constant does not change feasibility. These invariances motivate normalized observations and normalized value targets.

For repeated decision making across heterogeneous instances, we develop a theory-guided Deep FQI method. Fitted Q-iteration is a batch approximate dynamic-programming approach [11,12]; here, it is adapted to the deterministic acyclic structure of the polyomino-chain design problem. Separate horizon modules are trained backward, both feasible actions are evaluated through the exact model, and an auxiliary loss emphasizes the action-value gap that determines the greedy decision. Exact dynamic programming is kept outside the training loop and used only as an oracle for certification.

The main contributions are therefore the following:

- An exact three-dimensional reliability environment for sequential cost-aware design;
- Exact hard-budget structure, including budget saturation for homogeneous kink costs;
- An alpha-vector dynamic program with safe componentwise dominance pruning;
- An exact cost–alpha Pareto dynamic program for heterogeneous costs, simultaneously supporting hard- and soft-cost formulations;
- Proofs of reliability-state and cost-scale policy invariance;
- A proof that the ordered remaining cost suffix is required for a heterogeneous-cost Markov policy;

- A normalized, backward, full-action Deep FQI method with exact model transitions but no optimal-action labels;
- Exact-certified evaluation on unseen heterogeneous-cost instances with multiple independent training seeds and paired statistical comparisons.

The remainder of the paper is organized as follows. Section 2 records the exact reduced reliability dynamics and their relation to the full transfer representation. Section 3 develops homogeneous budget structure, alpha-vector pruning, and heterogeneous cost–alpha Pareto dynamic programming. Section 4 gives the sequential formulation and proves the policy invariances used for normalization. Section 5 introduces the theory-guided Deep FQI algorithm and its validation discipline. Section 6 presents exact-certified experiments and statistical comparisons, followed by discussion and conclusions.

2. Exact Reliability Dynamics

A chain with $n \geq 2$ squares has

$$m = n - 2$$

binary attachment decisions. An attachment word is

$$\omega = (\omega_1, \ldots, \omega_m) \in \{0,1\}^m,$$

where 0 denotes a linear attachment and 1 a kink. Every uncertain edge operates independently with probability $r \in (0,1)$, and $q = 1 - r$.

The exact reliability dynamics needed below are captured by the row state

$$\mathbf{x}_\omega = (b_\omega, e_\omega, c_\omega).$$

The initial state is

$$\mathbf{x}_\emptyset = \left(rq^3(1+2rq),\ r^3q^2(3-2r),\ r^2q^4\right).$$

Define

$$A(r) = \begin{pmatrix} rq(1+r) & r^3 & 0 \\ rq(2-r) & r^2(2-r) & rq^2 \\ rq & r^2 & rq \end{pmatrix}, \qquad J = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}.$$

Then

$$\mathbf{x}_{\omega 0} = \mathbf{x}_\omega A(r), \qquad \mathbf{x}_{\omega 1} = \mathbf{x}_\omega JA(r).$$

The two-terminal reliability is recovered exactly from

$$R_\omega(r) = \mathcal{R}(\mathbf{x}_\omega; r) = \frac{rb_\omega + e_\omega + c_\omega}{q^2}.$$

These recurrences are an invariant reduction of the full boundary-state transfer model [6]. In particular, if two prefixes have the same three-dimensional state, then every common continuation produces the same future state and terminal reliability. Thus $\mathbf{x}$ is reliability-sufficient for the sequential design problem.

For later use, put

$$B_0 = A(r), \qquad B_1 = JA(r),$$

so that for any continuation word $\gamma = (\gamma_1, \ldots, \gamma_h)$,

$$\mathbf{x}_{\omega\gamma} = \mathbf{x}_\omega B_\gamma, \qquad B_\gamma = B_{\gamma_1} \cdots B_{\gamma_h}.$$

All entries of B_0 and B_1 are nonnegative for $0 < r < 1$.

2.1. Relation to the Full Transfer Representation

For completeness, we briefly record how the three-dimensional state used in this paper is embedded in the exact boundary-state formulation. The original reliability model tracks the connectivity partition of the source and the three active boundary vertices, giving a fifteen-state probability vector. If $\mathbf{p}_2(r)$ denotes the exact initial boundary distribution and $T_0, T_1 \in \mathbb{R}^{15\times 15}$ are the attachment matrices, then

$$R_\omega(r) = \mathbf{p}_2 T_{\omega_1} \cdots T_{\omega_m} \mathbf{c}. \tag{1}$$

For this particular independent-edge, common-r model, the reachable fifteen-dimensional dynamics admit the stated five-coordinate invariant representation. The equality in (2.4) is an algebraic intertwining condition, not a consequence of dimension counting alone; its verification requires the full boundary-state matrices and the embedding Q. Writing

$$\mathbf{z}_\omega = (a_\omega, b_\omega, c_\omega, d_\omega, e_\omega),$$

the exact initial row vector is

$$\mathbf{z}_\emptyset = \big(q^5(1+2r),\ rq^3(1+2rq),\ r^2q^4,\ r^2q^4,\ r^3q^2(3-2r)\big). \tag{2}$$

The reduced attachment matrices are

$$M_0 = \begin{pmatrix} q(1+r) & 0 & 0 & r^2 & 0 \\ q^2(1+r) & rq(1+r) & 0 & r^2q & r^3 \\ q^2 & rq & rq & 0 & r^2 \\ q(1+r-r^2) & 0 & 0 & r^2(2-r) & 0 \\ q^3 & rq(2-r) & rq^2 & 0 & r^2(2-r) \end{pmatrix},$$

and

$$M_1 = \begin{pmatrix} q(1+r) & 0 & 0 & r^2 & 0 \\ q^2 & rq & rq & 0 & r^2 \\ q^2(1+r) & rq(1+r) & 0 & r^2q & r^3 \\ q(1+r-r^2) & 0 & 0 & r^2(2-r) & 0 \\ q^3 & rq(2-r) & rq^2 & 0 & r^2(2-r) \end{pmatrix}. \tag{3}$$

There exists an attachment-independent, parameter-dependent embedding matrix $Q(r)$ such that

$$\mathbf{p}_2 = \mathbf{z}_\emptyset Q, \qquad QT_i = M_i Q, \qquad i \in \{0,1\}, \tag{4}$$

and the reliability output satisfies

$$Q\mathbf{c} = \begin{pmatrix} 0 \\ r/q^2 \\ 1/q^2 \\ 0 \\ 1/q^2 \end{pmatrix}. \tag{5}$$

Consequently,

$$R_\omega(r) = \frac{rb_\omega + c_\omega + e_\omega}{q^2}. \tag{6}$$

The coordinates $(b_\omega, e_\omega, c_\omega)$ form a closed subsystem: with the row-vector convention used throughout this paper, they evolve exactly through A and JA as stated above. Thus the three-dimensional model is not a learned feature map or an empirical state compression; conditional on the intertwining identities Equation (4), it is an exact reliability-sufficient quotient of the transfer dynamics. The two actions preserve the three-coordinate subsystem because their updates depend only on these coordinates; the terminal output Equation (6)

also depends only on them. The reduction is specific to the stated graph-growth and edge-failure assumptions: it cannot be assumed for arbitrary extra boundary connections, altered terminal events, edge-specific reliability parameters, or transitions that couple the chosen coordinates to discarded ones. In such extensions, closure and output sufficiency must be established afresh from the full transfer model. This distinction is important later, because the learning algorithm approximates only the long-horizon decision value, never the network transition or reliability calculation.

2.2. Constructive Certification of the Fifteen-to-Three-State Reduction

To make the state reduction reproducible, we give an explicit intertwining map rather than relying on dimension counting. Fix $q = 1 - r$ and $0 < r < 1$. With the fifteen boundary partitions ordered as in the underlying transfer construction [7], define the 5×15 matrix $Q(r)$ by the row-linear identity

$$\begin{aligned}(a,b,c,d,e)Q(r) = \big(&a, b, 0, c, ra/q, d, ra/q, rb/q, 0, rc/q,\\ &rb/q, e, rc/q, [r^2a + r(2-r)d]/q^2,\\ &[r^2(b+c) + r(2-r)e]/q^2\big).\end{aligned}$$

Here the coordinates (a, b, c, d, e) are formal row-vector variables; $Q(r)$ is independent of the attachment word but depends on r. The fifteen states are the connectivity partitions of the ordered boundary $(s, 1, 2, 3)$. For each current partition, T_0 and T_1 are constructed by enumerating the eight operating/failed configurations of the three edges of the next square. For the linear action, the new edges are $(2, u)$, (u, v), $(v, 3)$ and the next active triple is $(2, u, v)$; for a kink, the new edges are $(1, u)$, (u, v), $(v, 2)$ and the next triple is $(2, v, u)$. Each configuration with j operating edges contributes $r^j q^{3-j}$ to the corresponding transition entry.

With M_0, M_1 as in (2.3), direct symbolic evaluation of those finite-state transitions gives

$$Q(r)T_i(r) = M_i(r)Q(r), \qquad i = 0, 1.$$

The seven initial ordinary edges, enumerated over all 2^7 operating/failed assignments with the two source links perfectly reliable, similarly give $\mathbf{p}_2 = \mathbf{z}_\emptyset Q(r)$. To see the final reduction explicitly, let

$$P = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \quad \mathbf{x}_\omega = \mathbf{z}_\omega P = (b_\omega, e_\omega, c_\omega).$$

A further exact multiplication yields

$$M_0 P = PA, \qquad M_1 P = PJA, \qquad Q(r)\mathbf{c} = P\begin{pmatrix} r/q^2 \\ 1/q^2 \\ 1/q^2 \end{pmatrix}.$$

Combining these identities proves, for *every* binary attachment word ω and not just those tested numerically,

$$\mathbf{p}_2 T_\omega \mathbf{c} = \mathbf{z}_\emptyset M_\omega Q(r)\mathbf{c} = \mathbf{x}_\emptyset B_\omega \begin{pmatrix} r/q^2 \\ 1/q^2 \\ 1/q^2 \end{pmatrix}.$$

The full symbolic verification script is provided in the accompanying computational audit material. This certifies the reduction under the specified boundary convention, independent and identically reliable ordinary edges, and perfectly reliable auxiliary terminal links; changed interface topology or nonidentical edge reliabilities require a new invariant-subspace proof.

2.3. Coordinate Recurrences

For reference, the two actions give the explicit updates

$$\begin{aligned} b_{\omega 0} &= rq(1+r)b_\omega + rq\,c_\omega + rq(2-r)e_\omega, \\ e_{\omega 0} &= r^3 b_\omega + r^2 c_\omega + r^2(2-r)e_\omega, \\ c_{\omega 0} &= rq\,c_\omega + rq^2 e_\omega, \end{aligned} \tag{7}$$

and

$$\begin{aligned} b_{\omega 1} &= rq\,b_\omega + rq(1+r)c_\omega + rq(2-r)e_\omega, \\ e_{\omega 1} &= r^2 b_\omega + r^3 c_\omega + r^2(2-r)e_\omega, \\ c_{\omega 1} &= rq\,b_\omega + rq^2 e_\omega. \end{aligned} \tag{8}$$

These formulas provide a useful independent implementation check on the matrix updates and make clear that every reachable coordinate remains nonnegative for $0 < r < 1$.

A structural result from the exact reliability analysis will be used throughout. Let

$$\delta_\omega = b_\omega - c_\omega > 0.$$

Then, for arbitrary binary prefixes and suffixes α, β,

$$R_{\alpha 1\beta}(r) - R_{\alpha 0\beta}(r) = \frac{\delta_\alpha \delta_{\beta^R}}{q^4} > 0. \tag{9}$$

Hence replacing a linear attachment by a kink at a fixed position, while keeping all other decisions unchanged, strictly increases reliability.

3. Cost-Constrained Exact Optimization

Let the incremental kink costs be

$$\kappa_t > 0, \qquad t = 1, \dots, m,$$

and define

$$C(\omega) = \sum_{t=1}^{m} \kappa_t \omega_t.$$

The hard-budget problem is

$$R^*(B; r) = \max_{\omega:\, C(\omega) \le B} R_\omega(r). \tag{10}$$

The complementary penalized problem is

$$J^*(\lambda; r) = \max_\omega \{R_\omega(r) - \lambda C(\omega)\}, \qquad \lambda \ge 0. \tag{11}$$

3.1. Homogeneous Cost and Budget Saturation

If $\kappa_t \equiv \kappa$, define

$$K(B) = \min\left(m, \left\lfloor \frac{B}{\kappa} \right\rfloor\right).$$

Proposition 1 (Budget saturation)**.** *For homogeneous positive kink cost, every optimizer of the hard-budget problem contains exactly $K(B)$ kink attachments.*

Proof. Suppose a feasible optimizer contained fewer than $K(B)$ ones. Since its length is m, at least one coordinate is zero, and changing that coordinate to one increases the cost by exactly κ. The resulting cost is at most $K(B)\kappa \le B$, so the modified design remains feasible. By the strict kink-improvement identity, its reliability is strictly larger, a contradiction. □

Therefore, with

$$F_K(r) = \max_{|\omega|_1=K} R_\omega(r),$$

we have

$$R^*(B;r) = F_{K(B)}(r).$$

The saturation statement does not imply uniqueness at fixed K.

For heterogeneous costs, exact budget equality need not hold because of discrete residual budget. However, any optimizer ω^* must satisfy the necessary condition

$$\omega_t^* = 0 \quad \Longrightarrow \quad C(\omega^*) + \kappa_t > B, \tag{12}$$

since otherwise changing that coordinate to a kink would remain feasible and would strictly increase reliability.

3.2. Alpha-Vector Representation

Write the reliability functional as

$$\mathcal{R}(\mathbf{x};r) = \mathbf{x}\boldsymbol{\nu}, \qquad \boldsymbol{\nu} = \begin{pmatrix} r/q^2 \\ 1/q^2 \\ 1/q^2 \end{pmatrix}.$$

For a continuation γ, define

$$\boldsymbol{\alpha}_\gamma = B_\gamma \boldsymbol{\nu}.$$

Then

$$\mathcal{R}(\mathbf{x}B_\gamma;r) = \mathbf{x}\boldsymbol{\alpha}_\gamma. \tag{13}$$

Thus every continuation is a linear functional of the current reliability state.

For fixed remaining horizon h and fixed cardinality K, let

$$\Gamma_{h,K} = \{\gamma \in \{0,1\}^h : |\gamma|_1 = K\}.$$

The exact continuation value is

$$W_{h,K}(\mathbf{x};r) = \max_{\gamma\in\Gamma_{h,K}} \mathbf{x}\boldsymbol{\alpha}_\gamma. \tag{14}$$

Proposition 2 (Componentwise dominance)**.** *If $\boldsymbol{\alpha} \le \boldsymbol{\beta}$ componentwise, then*

$$\mathbf{x}\boldsymbol{\alpha} \le \mathbf{x}\boldsymbol{\beta}$$

for every nonnegative state $\mathbf{x}$. Moreover, for $a \in \{0,1\}$,

$$B_a\boldsymbol{\alpha} \le B_a\boldsymbol{\beta}.$$

Proof. The first assertion follows from $\mathbf{x}(\boldsymbol{\beta} - \boldsymbol{\alpha}) \ge 0$. The second follows because B_0 and B_1 are entrywise nonnegative. □

Let ND remove componentwise dominated vectors. The exact nondominated frontier therefore satisfies

$$\mathcal{F}_{0,0} = \{\boldsymbol{\nu}\},$$

and, for $h \ge 1$,

$$\mathcal{F}_{h,K} = \mathrm{ND}\big(B_0\mathcal{F}_{h-1,K} \cup B_1\mathcal{F}_{h-1,K-1}\big), \tag{15}$$

with infeasible indices interpreted as the empty set. Induction using componentwise dominance proves

$$W_{h,K}(\mathbf{x};r) = \max_{\boldsymbol{\alpha}\in\mathcal{F}_{h,K}} \mathbf{x}\boldsymbol{\alpha}. \tag{16}$$

No polynomial worst-case bound for $|\mathcal{F}_{h,K}|$ is assumed; the gain depends on the amount of dominance in the particular problem.

3.3. Cost–Alpha Pareto Dynamic Programming

The heterogeneous problem can be handled without discretizing the budget. Associate with each continuation the pair

$$\Pi_\gamma = (C(\gamma), \boldsymbol{\alpha}_\gamma).$$

We say that $(c_1, \boldsymbol{\alpha}_1)$ dominates $(c_2, \boldsymbol{\alpha}_2)$ if

$$c_1 \le c_2, \qquad \boldsymbol{\alpha}_1 \ge \boldsymbol{\alpha}_2$$

componentwise, with at least one strict inequality.

Theorem 1 (Exact Cost–Alpha Frontier)**.** *Let $\mathcal{P}_t$ be the nondominated cost–alpha pairs for continuations beginning at position t. With*

$$\mathcal{P}_{m+1} = \{(0, \boldsymbol{\nu})\},$$

define

$$C_t = \{(c, B_0\boldsymbol{\alpha}) : (c, \boldsymbol{\alpha}) \in \mathcal{P}_{t+1}\} \cup \{(c + \kappa_t, B_1\boldsymbol{\alpha}) : (c, \boldsymbol{\alpha}) \in \mathcal{P}_{t+1}\},$$

and let $\mathcal{P}_t$ be obtained by deleting all cost–alpha dominated pairs from C_t. Then, for every nonnegative state $\mathbf{x}$ and every budget B,

$$R^*(B; \mathbf{x}, r) = \max_{(c,\boldsymbol{\alpha}) \in \mathcal{P}_t,\ c \le B} \mathbf{x}\boldsymbol{\alpha}. \tag{17}$$

The same frontier also gives, for every $\lambda \ge 0$,

$$J^*(\lambda; \mathbf{x}, r) = \max_{(c,\boldsymbol{\alpha}) \in \mathcal{P}_t} (\mathbf{x}\boldsymbol{\alpha} - \lambda c). \tag{18}$$

Proof. If $(c_1, \boldsymbol{\alpha}_1)$ dominates $(c_2, \boldsymbol{\alpha}_2)$, then whenever the second continuation is budget-feasible the first is also feasible, and for every nonnegative state $\mathbf{x}\boldsymbol{\alpha}_1 \ge \mathbf{x}\boldsymbol{\alpha}_2$. The dominance relation is preserved when the same action is prepended, because the same nonnegative action cost is added and B_0, B_1 are entrywise nonnegative. Backward induction therefore removes no pair that can be optimal for any state or budget, proving the hard-budget frontier formula. For $\lambda \ge 0$, dominance also implies $\mathbf{x}\boldsymbol{\alpha}_1 - \lambda c_1 \ge \mathbf{x}\boldsymbol{\alpha}_2 - \lambda c_2$, yielding the penalized frontier formula. □

The frontier size can still be exponential in the worst case. With a direct pairwise dominance test and $N_t = |\mathcal{P}_t|$, a simple implementation has pruning cost of order

$$O\left(\sum_t N_t^2\right)$$

and memory of order $O(\max_t N_t)$, up to constant factors. The method is therefore an exact dominance-pruned dynamic program, not a claimed polynomial-time algorithm.

3.4. Geometry of the Exact Value Function

The alpha representation also gives a useful geometric description of the optimization problem. For fixed h, K, r, every admissible continuation induces the linear functional

$$\mathbf{x} \longmapsto \mathbf{x}\boldsymbol{\alpha}_\gamma.$$

Hence

$$W_{h,K}(\mathbf{x}; r) = \max_{\gamma \in \Gamma_{h,K}} \mathbf{x}\boldsymbol{\alpha}_\gamma \tag{19}$$

is a pointwise maximum of finitely many linear functions.

Corollary 1 (Piecewise-linear convexity)**.** *For fixed h, K, r, the exact value function $W_{h,K}$ is convex, piecewise linear, and positively homogeneous on the nonnegative state cone. Each nondominated alpha-vector determines a candidate linear region of the value function.*

Proof. A finite maximum of linear functions is convex and piecewise linear. For $c > 0$,

$$W_{h,K}(c\mathbf{x}; r) = \max_{\gamma} c\mathbf{x}\boldsymbol{\alpha}_{\gamma} = cW_{h,K}(\mathbf{x}; r).$$

□

This observation has two roles. First, it explains why alpha-vector pruning is an exact value-function compression rather than merely a word-space heuristic. Second, it motivates the structure-aware normalization used by the learned value approximator in Section 5.

3.5. Exactness and Computational Scope

The exact and learned methods address different computational regimes. For a single fixed instance, the cost–alpha frontier is preferable whenever it can be constructed within the available computational budget: it gives a global certificate and simultaneously answers all hard-budget and nonnegative soft-penalty queries for the same $(r, \boldsymbol{\kappa})$ profile. The frontier size, however, is data-dependent. In the absence of a polynomial upper bound on $N_t = |\mathcal{P}_t|$, exact computation may still become costly for long heterogeneous profiles or for profiles with weak dominance.

For comparison, direct evaluation of one specified attachment word costs only $O(m)$ because the reliability state has fixed dimension. The combinatorial difficulty lies in design selection. Unrestricted exhaustive search evaluates 2^m words, while homogeneous fixed-cardinality search contains $\binom{m}{K}$ candidates. At the hardest cardinalities,

$$\binom{m}{\lfloor m/2 \rfloor} \sim \frac{2^m}{\sqrt{\pi m/2}}. \tag{20}$$

The purpose of dominance pruning is to replace these raw candidate counts by the sizes of the exact nondominated frontiers, not to alter the objective.

4. Sequential Decision Structure and Invariances

At stage k, let $\mathbf{x}_k$ be the exact reliability state, B_k the remaining budget, and $a_k \in \{0, 1\}$ the next attachment. The transition is

$$\mathbf{x}_{k+1} = \mathbf{x}_k B_{a_k}, \qquad B_{k+1} = B_k - \kappa_{k+1} a_k,$$

with feasible action set

$$\mathcal{A}_k(B_k) = \begin{cases} \{0, 1\}, & B_k \geq \kappa_{k+1}, \\ \{0\}, & B_k < \kappa_{k+1}. \end{cases}$$

For the hard-budget problem define the shaped reward

$$\rho_k = \mathcal{R}(\mathbf{x}_{k+1}; r) - \mathcal{R}(\mathbf{x}_k; r). \tag{21}$$

Proposition 3 (Reward equivalence)**.** *For every feasible attachment word,*

$$\sum_{k=0}^{m-1} \rho_k = R_{\omega}(r) - R_{\emptyset}(r). \tag{22}$$

Consequently, with undiscounted return ($\gamma = 1$), maximizing cumulative reward is exactly equivalent to maximizing terminal reliability under the budget constraint.

Proof. The displayed reward identity is the telescoping sum of successive reliability differences. The initial reliability is independent of the policy. □

For the penalized objective, replacing the hard-budget reward by

$$\rho_k^{\lambda} = \mathcal{R}(\mathbf{x}_{k+1}; r) - \mathcal{R}(\mathbf{x}_k; r) - \lambda \kappa_{k+1} a_k$$

yields cumulative return $R_\omega - \lambda C(\omega) - R_\emptyset$.

4.1. Finite-Horizon Bellman Equations

For a fixed heterogeneous instance, let $V_k(\mathbf{x}, B)$ denote the optimal cumulative shaped reward from stage k when the current reliability state is $\mathbf{x}$ and the remaining budget is B. With $V_m \equiv 0$, the hard-budget Bellman recursion is

$$V_k(\mathbf{x}, B) = \max_{a \in A_k(B)} \left\{ \mathcal{R}(\mathcal{T}_a(\mathbf{x}); r) - \mathcal{R}(\mathbf{x}; r) + V_{k+1}\big(\mathcal{T}_a(\mathbf{x}), B - \kappa_{k+1} a\big) \right\}. \tag{23}$$

The associated action value is

$$Q_k(\mathbf{x}, B, a) = \mathcal{R}(\mathcal{T}_a(\mathbf{x}); r) - \mathcal{R}(\mathbf{x}; r) + V_{k+1}\big(\mathcal{T}_a(\mathbf{x}), B - \kappa_{k+1} a\big). \tag{24}$$

For the soft-cost formulation the same recursion holds after subtracting $\lambda \kappa_{k+1} a$ from the immediate reward. No discounting is introduced: $\gamma = 1$ is required if the Bellman return is to remain exactly equivalent to the terminal design objective.

The Bellman formulation also clarifies why a myopic strategy is generally insufficient. Equation (9) shows that, if all other decisions are held fixed, replacing a zero by a one increases reliability. Under a budget, however, taking a kink now can remove a more valuable kink opportunity later. Thus the local monotonicity theorem does not imply a greedy allocation rule. The continuation value in Equation (23) is precisely the term that represents this intertemporal opportunity cost.

4.2. Fixed-Instance and Generalized Markov States

For one fixed instance the position k identifies the future costs, so $(k, \mathbf{x}_k, B_k)$ is a sufficient state. This is no longer true for a single policy shared across different heterogeneous profiles. Such a policy must distinguish states that have identical reliability coordinates and remaining budget but different future prices. A fully informative generalized state can therefore be written as

$$s_k^{\text{het}} = (\mathbf{x}_k, B_k, r, \kappa_{k+1}, \ldots, \kappa_m). \tag{25}$$

At any decision, the state transition removes the first suffix entry, updates the remaining budget by the chosen action cost, and propagates the three-dimensional reliability state. Hence the augmented state together with the remaining horizon is Markov for the deterministic design problem. The suffix is fixed problem data for any single instance but must be present in the observation of a policy intended to operate across varying instances. In the absence of restrictions on the heterogeneous cost profiles, no universal fixed-dimensional summary is established here: the counterexample below excludes budget, horizon, and current cost alone, but does not prove that every imaginable lower-dimensional encoding is impossible. The following subsections reduce this representation only by transformations that are proved to preserve the optimal policy.

4.3. Reliability-State Scale Invariance

The exact transition and reliability functional are linear. Therefore, for every $c > 0$,

$$\mathcal{T}_a(c\mathbf{x}) = c\mathcal{T}_a(\mathbf{x}), \qquad \mathcal{R}(c\mathbf{x}; r) = c\mathcal{R}(\mathbf{x}; r).$$

It follows immediately from the fixed-cardinality value representation that

$$W_{h,K}(c\mathbf{x}; r) = cW_{h,K}(\mathbf{x}; r). \tag{26}$$

Thus positive rescaling of the reliability state does not change the set of optimal continuations.

Let

$$S(\mathbf{x}) = x_1 + x_2 + x_3 > 0, \qquad \bar{\mathbf{x}} = \frac{\mathbf{x}}{S(\mathbf{x})}.$$

Then

$$W_{h,K}(\mathbf{x}; r) = S(\mathbf{x}) W_{h,K}(\bar{\mathbf{x}}; r),$$

so the policy is determined by the normalized state direction together with the remaining decision information. This also explains why raw Q-values can become numerically small with increasing horizon even when the action ordering remains meaningful.

For a normalized action value

$$\widetilde{Q} = Q/S(\mathbf{x}),$$

let $\mathbf{x}^{(a)} = \overset{\frown}{J}_a(\mathbf{x})$ and

$$\sigma_a = \frac{S(\mathbf{x}^{(a)})}{S(\mathbf{x})}.$$

Then the normalized Bellman recursion has the form

$$\widetilde{Q}_h(\bar{\mathbf{x}}, a) = \tilde{\rho}(\bar{\mathbf{x}}, a) + \sigma_a \max_{a'} \widetilde{Q}_{h-1}(\bar{\mathbf{x}}^{(a)}, a'), \tag{27}$$

where

$$\tilde{\rho}(\bar{\mathbf{x}}, a) = \mathcal{R}(\overset{\frown}{J}_a(\bar{\mathbf{x}}); r) - \mathcal{R}(\bar{\mathbf{x}}; r).$$

Dividing both feasible action values by the same positive scale is policy-preserving.

4.4. Why the Heterogeneous Cost Suffix Is Part of the State

For a policy shared across different heterogeneous cost profiles, $(\mathbf{x}, B, h, r)$ is not Markov. Even adding only the current cost is insufficient. Consider

$$r = \frac{1}{2}, \quad m = 3, \quad B = 2, \quad \kappa_1 = 2.$$

For the profile

$$(\kappa_1, \kappa_2, \kappa_3) = (2, 1, 1),$$

the unique optimal word is

$$011, \qquad R_{011}(1/2) = \frac{1635}{8192} \approx 0.1995849609,$$

so the optimal first action is 0. With the same current state, budget, horizon, r, and current cost, but future profile

$$(2, 1, 3),$$

the optimum is

$$100, \qquad R_{100}(1/2) = \frac{373}{2048} \approx 0.1821289063,$$

whereas the best feasible continuation beginning with zero is 010, with

$$R_{010}(1/2) = \frac{745}{4096} \approx 0.1818847656.$$

The optimal first action is therefore 1. Hence the ordered remaining cost suffix is decision-relevant.

4.5. Cost-Scale Invariance

If the remaining budget and every remaining cost are multiplied by the same constant $c > 0$, then

$$\sum_j \kappa_j \gamma_j \le B \quad \Longleftrightarrow \quad \sum_j c\kappa_j \gamma_j \le cB.$$

The feasible continuation set, and therefore the hard-budget optimal policy, is unchanged.

For a remaining suffix $(\kappa_{k+1}, \dots, \kappa_m)$ define

$$D_k = B_k + \sum_{j=k+1}^{m} \kappa_j, \qquad \beta_k = \frac{B_k}{D_k}, \qquad \hat{\kappa}_{k,j} = \frac{\kappa_{k+j}}{D_k}. \tag{28}$$

For nonterminal states, $D_k > 0$ and

$$\beta_k + \sum_{j=1}^{h_k} \hat{\kappa}_{k,j} = 1.$$

Thus $(\beta_k, \hat{\boldsymbol{\kappa}}^{(k)})$ retains the complete ordered remaining cost profile up to a policy-irrelevant common scale.

Proposition 4 (Scale-free heterogeneous policy state)**.** *For a bounded horizon, the optimal hard-budget action is determined by*

$$\left(\bar{\mathbf{x}}_k, r, \beta_k, \hat{\kappa}_{k,1}, \dots, \hat{\kappa}_{k,h_k}\right), \tag{29}$$

together with the remaining horizon h_k. Positive rescaling of either the reliability state or the common monetary unit leaves the set of optimal actions unchanged.

Proof. Positive reliability-state rescaling multiplies the value of every feasible continuation by the same constant, while common cost–budget rescaling leaves the feasible continuation set unchanged. The normalized variables retain the complete state direction and the ordered cost suffix modulo precisely these two policy-irrelevant scales. □

The result is stronger than ordinary feature normalization: it identifies an exact quotient of the decision state. In particular, no empirical summary such as mean future cost, minimum cost, or cost variance can replace the ordered suffix in the principal formulation, because the counterexample above shows that order can reverse the optimal action.

5. Theory-Guided Deep Fitted Q-Iteration

The exact cost–alpha dynamic program is preferred whenever an exact solution for a particular instance is computationally affordable. The role of learning is different: to amortize decision making across many previously unseen heterogeneous instances without rebuilding an exact frontier for every new cost profile.

Because the environment is deterministic, the action set is binary, and the horizon is finite and acyclic, we use backward fitted Q-iteration rather than generic online DQN. Separate networks $\widetilde{Q}_h$, $h = 1, \dots, m_{\mathrm{ref}}$, are associated with the number of decisions remaining. The horizon is therefore encoded by the choice of module and need not be repeated as an input feature.

For a module with h decisions remaining, the input retains the complete normalized suffix and therefore has dimension $h + 5$:

$$o_h^{\mathrm{het}} = (\bar{x}_1, \bar{x}_2, \bar{x}_3,\ r,\ \beta, \hat{\kappa}_1, \dots, \hat{\kappa}_h) \in \mathbb{R}^{h+5}. \tag{30}$$

No zero padding is required because a separate module is used for each remaining horizon. Thus each horizon module is a fully connected network

$$(h+5) \longrightarrow 64 \longrightarrow 64 \longrightarrow 2$$

with ReLU hidden units and two linear action outputs. For a sampled state, the exact model generates the successor and target for every feasible action. At horizon $h = 1$,

$$y_1(s,a) = \tilde{\rho}(s,a).$$

After fitting $\widetilde{Q}_1$, its parameters are frozen. For $h > 1$,

$$y_h(s,a) = \tilde{\rho}(s,a) + \sigma_a \max_{a' \in \bar{A}(s')} \widetilde{Q}_{h-1}(o',a'). \tag{31}$$

Training therefore proceeds as

$$\widetilde{Q}_1 \to \widetilde{Q}_2 \to \cdots \to \widetilde{Q}_{m_{\text{ref}}}.$$

No exact optimal word or exact Bellman value is used to construct these training targets.

At branching states define

$$\Delta y = y(s,1) - y(s,0), \qquad \Delta\widetilde{Q} = \widetilde{Q}(s,1) - \widetilde{Q}(s,0).$$

The training loss is

$$\mathcal{L} = \mathcal{L}_Q + \mu \mathcal{L}_{\text{gap}}, \qquad \mu = 2, \tag{32}$$

where

$$\mathcal{L}_Q = \frac{1}{N_{\text{feas}}} \sum_{s,a \in \bar{A}(s)} (\widetilde{Q}(s,a) - y(s,a))^2$$

and

$$\mathcal{L}_{\text{gap}} = \frac{1}{N_{\text{br}}} \sum_{s:\,|\bar{A}(s)|=2} (\Delta\widetilde{Q}(s) - \Delta y(s))^2.$$

The auxiliary term does not introduce oracle labels; it only emphasizes the relative Bellman targets of the two exactly known action transitions.

5.1. Decision Sensitivity to Action-Value Approximation

Let $Q_h^*(s,a)$ be the true optimal remaining-horizon action value, with the same normalization as the fitted values. Suppose both actions are feasible and $|\widetilde{Q}_h(s,a) - Q_h^*(s,a)| \le \varepsilon$ for each. If a^* maximizes Q_h^* and $\hat{a}$ maximizes $\widetilde{Q}_h$, then

$$0 \le Q_h^*(s,a^*) - Q_h^*(s,\hat{a}) \le 2\varepsilon.$$

Indeed, insert the approximate values between the exact values and use the greedy optimality of $\hat{a}$. Therefore a strict exact action gap exceeding 2ε guarantees that the greedy action cannot be reversed. More generally, an action reversal can occur only in states where the exact action gap is no larger than 2ε. This is a *conditional* deterministic error bound, not a learned global error certificate: training loss on sampled states does not establish a uniform ε on unseen states. The exact oracle is therefore used only at validation or test states to measure actual action gaps and optimality losses. The action-gap loss in Equation (32) is motivated by this sensitivity but does not guarantee that the bound's premise holds.

5.2. Why Backward Fitted Q-Iteration Is Used

A generic online DQN would ignore several favorable features of the present problem. The transition model is known exactly, the action set has only two members, and the decision graph is acyclic with a fixed finite horizon. In addition, pilot calculations show that two raw action values can share a large common component while their decision-relevant difference is much smaller. Approximating both raw values directly can therefore produce the correct scale but the wrong action ordering.

The proposed construction addresses these issues in three ways. First, positive homogeneity removes the irrelevant reliability magnitude. Second, full-action model evaluation supplies a Bellman target for every feasible action at a sampled state, so no transition information is discarded. Third, horizon-wise backward fitting freezes the entire lower-horizon target module before the next module is trained. This eliminates the moving-target interaction between online and target networks that motivates many of the stabilization devices used in generic DQN.

The exact cost–alpha oracle is absent from Algorithm 1. Training uses only the mathematical environment and previously fitted horizon modules. Exact solutions are introduced later solely for validation and test certification.

Algorithm 1 Theory-guided finite-horizon Deep FQI

Require: Maximum horizon m_{ref}; balanced training-state sets $\mathcal{D}_h$, $h = 1, \dots, m_{\mathrm{ref}}$

for $h = 1$ to m_{ref} **do**
  for each sampled state $s \in \mathcal{D}_h$ **do**
    Construct the scale-free observation $o(s)$
    for each feasible action $a \in \mathcal{A}(s)$ **do**
      Compute the exact successor $s' = T(s, a)$ and normalized reward $\tilde{\rho}(s, a)$
      if $h = 1$ **then**
        $y(s, a) \leftarrow \tilde{\rho}(s, a)$
      else
        $y(s, a) \leftarrow \tilde{\rho}(s, a) + \sigma_a \max_{a' \in \mathcal{A}(s')} \widetilde{Q}_{h-1}(o(s'), a')$
      end if
    end for
  end for
  Fit $\widetilde{Q}_h$ using $\mathcal{L}_Q + \mu \mathcal{L}_{\mathrm{gap}}$
  Freeze $\widetilde{Q}_h$
end for
return $\{\widetilde{Q}_1, \dots, \widetilde{Q}_{m_{\mathrm{ref}}}\}$

5.3. Training Protocol

Training states were balanced by remaining horizon: 3000 independently generated states were used for every $h = 1, \dots, 8$. The reliability parameter was sampled from the fixed mixture

$$0.7\, U(0.30, 0.75) + 0.3\, U(0.75, 0.90), \tag{33}$$

selected on a separate high-r validation set before final testing. For each state, the reliability component was obtained from a random reachable prefix; the remaining budget was sampled separately for the remaining cost suffix, so the full training state was not necessarily induced by a single feasible prefix budget trajectory. Half of the remaining cost profiles had independent costs from $U(0.5, 2.0)$; the other half were bimodal, with each position independently drawn from a low interval $U(0.3, 0.8)$ or a high interval $U(1.8, 3.0)$. The remaining budget was sampled as a fraction from $U(0.15, 0.80)$ of the sum of the remaining costs.

All final networks used Adam with learning rate 10^{-3}, weight decay 10^{-6}, batch size 256, at most 120 epochs per horizon, and patience 20. Five independent final training seeds,

$$101, 102, 103, 104, 105,$$

were fixed before evaluation.

5.3.1. Implementation-Level Checkpoint Selection

Within each horizon-specific fitted network, the archived implementation retains the checkpoint minimizing the epoch-aggregated training objective $\mathcal{L}_Q + 2\mathcal{L}_{\text{gap}}$; it stops after 20 consecutive epochs without an improvement exceeding 10^{-9}, or at 120 epochs. This is training-loss-based stopping, *not* early stopping on a held-out validation set. The separate high-r validation family mentioned below was used for the distribution/model-selection stage, not for selecting neural-network epochs. This distinction is important when interpreting generalization and potential overfitting. The archived code was run using Python 3.13.5 with the following dependencies: NumPy 2.3.5, pandas 2.2.3, SciPy 1.17.0, and PyTorch 2.10.0+cpu; the original requirements list does not pin package versions, which limits exact environment replication. The archived experimental CSVs permit independent checking of the reported numerical summaries. However, the original computational environment was not version-pinned, and the full five-seed neural-training trajectories have not been independently replayed in the present audit. Therefore, bitwise reproducibility across environments is not claimed.

5.3.2. Reconstructed Test-Instance Provenance and Independent Oracle Verification

The test-instance generator is deterministic conditional on its recorded NumPy random seeds: 9101 for the principal family (60 instances), 9201 for the high-r family (60 instances), and 9301 for the extrapolation family (40 instances). Each instance contains eight position-dependent kink costs, a reliability parameter r, and a hard budget B. Although the original summary CSVs did not explicitly retain the eight-dimensional cost profile, the profiles were subsequently reconstructed from the archived generator and these fixed seeds. The reconstructed r and B values agree with the original oracle archive up to 1.11×10^{-16} and 8.88×10^{-16}, respectively. Independent reruns of all 160 exact cost–alpha solves, all 160 brute-force optima, both feasible baselines, and the Pareto frontier sizes reproduce the archived results: the largest absolute discrepancy among the reported reliability values is 1.12×10^{-16}, and all frontier sizes agree exactly. These verifications establish reproducibility of the finite-horizon exact evaluation and baselines, but they do not constitute a new end-to-end retraining of the five Deep FQI policies. The reconstructed profiles, verification tables, and generation/solver scripts accompany the computational data package. The original archived policy predictions and training summaries are retained without alteration.

The experiments are entirely computational. Reliability parameters, position-dependent kink costs, and budgets are synthetically generated by the distributions specified above and in Table 1; the reachable reliability state is generated by applying exact transitions to a sampled prefix. There are no measurements, human participants, or external observational datasets. Possible sources of future field data and the necessary validation checks are described in Section 7.5. For auditability the retained result files include the 160-instance exact-oracle table, five-seed test predictions, per-seed and final summaries, and per-horizon training statistics. The five seeds are 101–105. The archived outputs support independent recalculation of the reported summary values; reproducing the whole training trajectory requires executing the accompanying source code in a compatible computational environment.

The exact cost–alpha solver was not used during training. It was used only to verify the environment and to certify final optimality gaps. This separation avoids leakage of exact optimal actions into the function approximator.

Table 1. Final train/validation/test roles. All final test cost profiles are unseen during training and validation.

Role	Number/Sampling	r Range	Purpose
Training	3000 states per horizon	mixture in (5.4)	Bellman fitting
Validation	separate high-r family	validation range	model selection
Principal test	60 instances	$[0.30, 0.75]$	in-range heterogeneous test
High-r test	60 instances	$[0.80, 0.88]$	difficult high-reliability regime
Extrapolation	40 instances	$[0.92, 0.96]$	out-of-range r

5.4. Validation and Model-Selection Discipline

According to the original experimental protocol, the high-r mixture in (5.4) was selected using a validation family separate from the final test instances, before the reported test evaluation. This family was used to compare reliability-parameter sampling choices, not to select individual training-epoch checkpoints. The archived final-training implementation does not include the exploratory model-selection runs; consequently, that historical validation comparison cannot be independently reconstructed from the supplied archive alone. The reported test results should be interpreted in light of this reproducibility limitation.

Five final training seeds were used to measure optimization variability. The same test instances were evaluated under each trained policy, but statistical comparisons against deterministic baselines were performed at the instance level after averaging the five learned-policy gaps. This prevents the five seeds for one physical test instance from being treated as five independent experimental units.

6. Exact-Certified Experiments

Three disjoint final test families were generated after the training distribution had been fixed. All have $m = 8$ and unseen heterogeneous cost profiles. The principal family contains 60 instances with $r \in [0.30, 0.75]$; the high-r family contains 60 instances with $r \in [0.80, 0.88]$; and the reliability-parameter extrapolation family contains 40 instances with $r \in [0.92, 0.96]$. Half of each family uses smooth heterogeneous costs sampled from $U(0.5, 2.0)$ and half uses bimodal low/high costs as above. Test budgets are sampled between 25% and 65% of total kink cost.

6.1. Evaluation Metrics and Baselines

For every test instance, the exact optimum R^* is certified independently. The primary metric is the absolute reliability gap

$$\Delta R(\pi) = R^* - R^\pi. \tag{34}$$

Absolute rather than relative gap is primary because R^* can become small in some reliability regimes, making percentage errors unstable. We also report exact-design recovery rate, the fraction of instances on which the seed-averaged learned gap is smaller than a baseline gap, paired Wilcoxon tests, and paired bootstrap confidence intervals for mean gap improvement.

Two inexpensive feasible baselines are used. The myopic policy selects the feasible action that maximizes one-step reliability after the current attachment. The random baseline chooses uniformly from the currently feasible actions; because $m = 8$ in the final experiments, its expected terminal reliability is evaluated exactly by enumerating the policy's feasible

random branches rather than estimated from a small Monte Carlo sample. These baselines deliberately represent low-computation policies rather than competitors to the exact oracle.

6.2. Independent Exact-Oracle Verification

Every one of the 160 final test instances was solved by both direct brute-force enumeration and cost–alpha Pareto dynamic programming. The maximum absolute discrepancy was

$$6.66 \times 10^{-16},$$

and the mean discrepancy was approximately

$$1.20 \times 10^{-16}.$$

Thus the reported optimality gaps are cross-checked against a separate finite enumeration, to the stated floating-point tolerance; this is not a formally verified interval-arithmetic certificate.

On 75 further small instances with randomly generated heterogeneous costs, the largest difference between cost–alpha optimization and brute-force enumeration was 2.22×10^{-16}.

Across the 160 final instances, the retained terminal cost–alpha frontier had mean size 76.19 and maximum size 124. These counts are far below the $2^8 = 256$ raw words, but they should not be interpreted as a universal compression factor: frontier size is problem dependent and may grow much more rapidly at longer horizons.

As a small-horizon runtime sanity check, the current Python implementation required approximately 3.4×10^{-3} seconds per instance for direct brute-force exact search, whereas the unoptimized cost–alpha implementation required about 9.9×10^{-2} seconds per instance because pairwise Pareto maintenance dominates at $m = 8$. Thus the exact Pareto method is introduced for its structural compression and multi-budget/multi-λ capability, not because it is faster than brute force at every small horizon.

6.3. Final Performance

Table 2 reports the mean absolute gap. For Deep FQI, the mean is computed for every training seed over the corresponding test family; the displayed standard deviation is the standard deviation of these five seed-level mean gaps.

Table 2. Exact-certified performance on unseen heterogeneous-cost test families.

Family	N	Deep FQI Gap	Myopic Gap	Random Gap	Red. vs. Myopic	Red. vs. Random
Principal	60	0.008870 ± 0.000875	0.013014	0.017574	31.84%	49.53%
High-r	60	0.008152 ± 0.000635	0.015686	0.027789	48.03%	70.67%
r-extrapolation	40	0.001124 ± 0.000092	0.004045	0.005988	72.22%	81.24%

For practical interpretation, the mean absolute reliability gap measures foregone connection probability, not a monetary saving or a guaranteed relative improvement for any individual network. For example, the reduction in the principal-family mean gap from 0.013014 (myopic) to 0.008870 (Deep FQI) corresponds to about 0.004144 in absolute two-terminal connection probability. In repeated deployments across new cost profiles, an amortized policy may avoid resolving a separate exact optimization problem at each decision; the claim here is confined to the tested synthetic distributions and the reported implementation. Where certified optimal design is mandatory, the Pareto solver remains preferable.

The exact-design recovery rates, pooled across five independent training seeds in each family, were 5.67%, 20.67%, and 27.50% on the principal, high-r, and extrapolation families, respectively. Hence the learned method is not an exact optimizer. Its benefit is the reduction of suboptimality across previously unseen instances.

After averaging the five learned-policy gaps at the instance level, Deep FQI produced a smaller gap than myopic on 61.67%, 68.33%, and 70.00% of the three test families, respectively. Against the exact expected random-feasible baseline, the corresponding proportions were 90.00%, 98.33%, and 100.00%.

6.4. Paired Statistical Comparisons

To avoid pseudo-replication, the five training-seed results were first averaged within each test instance. One-sided paired Wilcoxon signed-rank tests were then applied across independent test instances, with the alternative that Deep FQI has smaller gap. For numerical stability, paired differences of absolute magnitude below 10^{-12} were set to zero before ranking, consistently across all families and baselines (the usual Wilcoxon zero-exclusion convention was used). Table 3 reports the results. Paired bootstrap confidence intervals use 10,000 resamples of test instances.

Table 3. Paired instance-level improvements of Deep FQI. Positive mean improvement denotes a smaller optimality gap.

Family	Improvement vs. Myopic	95% CI	p	Improvement vs. Random	95% CI	p
Principal	0.004144	[0.001304, 0.007091]	0.003824	0.008705	[0.006224, 0.011330]	2.15×10^{-10}
High-r	0.007535	[0.003418, 0.011867]	0.001118	0.019637	[0.017003, 0.022377]	9.02×10^{-12}
r-extrapolation	0.002922	[0.001840, 0.004050]	7.0×10^{-6}	0.004865	[0.004192, 0.005564]	9.09×10^{-13}

All bootstrap intervals lie strictly above zero, and all paired tests reject equality in favor of smaller Deep-FQI gaps at the 0.01 level. The evidence supports a reproducible improvement over the two feasible baselines on the tested distributions. It does not establish global optimality of the learned policy.

6.5. Seed Stability and Computational Cost

The seed-to-seed standard deviations in Table 2 are smaller than the corresponding baseline gaps. This descriptive comparison shows variation across the five recorded initializations; it does not establish stability under other architectures, seeds, or data-generating distributions. The five archived final training runs were reported to require approximately 30–33 s each in the original CPU-based implementation. Post-training greedy inference on the length-eight collection required approximately

$$3.7 \times 10^{-4} \text{ s per instance,}$$

compared with about 6.0×10^{-5} seconds for the myopic policy and 3.4×10^{-3} seconds for direct brute-force exact search. These timings are implementation specific and are reported only to clarify the computational roles of the methods.

The learned method therefore occupies an intermediate regime: it is slower than a one-step greedy rule but returns substantially smaller gaps, while its post-training inference is faster than resolving the small exact problem from scratch. At larger horizons the relevant comparison is expected to depend on frontier growth, model architecture, and the number of new instances over which training cost is amortized.

7. Discussion

The results support a deliberately two-level interpretation of the proposed framework. Exact Pareto dynamic programming and Deep FQI solve related but not identical computational problems. For a particular $(r, B, \boldsymbol{\kappa})$ instance, exact optimization is the preferred method whenever the nondominated frontier can be constructed at acceptable cost. It

returns a global certificate and, for the same cost profile and reliability parameter, supports all subsequent hard-budget and nonnegative soft-penalty queries without rebuilding the continuation representation.

The learning problem is instead an amortization problem. A horizon-wise Deep FQI model is trained once on a distribution of reachable reliability states, budgets, reliability parameters, and ordered heterogeneous cost suffixes. After training, a new cost profile can be acted on directly without solving an exact Pareto recursion for that profile. The experiments therefore do not attempt to establish that learning is intrinsically superior to exact dynamic programming. They ask whether a reusable approximate policy can improve substantially on very inexpensive feasible rules while remaining independent of exact optimal-action labels.

7.1. Interpretation of the Exact Contribution

The alpha-vector construction is more than an implementation device. It identifies the exact finite family of linear continuation functionals that can matter at a state. Componentwise dominance is safe on the entire nonnegative state cone, and nonnegative transition matrices preserve this partial order backward through the construction. Introducing heterogeneous cost adds one coordinate to the dominance relation and produces a budget-independent Pareto frontier. This explains why a single exact object simultaneously answers hard-budget, soft-penalty, and cost–reliability trade-off queries.

The empirical frontier sizes observed here are encouraging but should be read cautiously. Small retained frontiers on the present instances do not imply a polynomial-time algorithm. In a weak-dominance regime the nondominated set can still become large. The exact contribution is therefore the correctness of the pruning representation and the resulting structural compression, not a worst-case complexity claim.

7.2. Why a Generic DQN Is Not the Principal Method

The final learning algorithm emerged from the mathematics of the exact model. In a raw action-value formulation, both actions may share a relatively large common continuation component while their difference is much smaller. A small absolute approximation error can then reverse the action ordering even when the two raw values appear numerically accurate. The positive-homogeneity result removes the irrelevant reliability magnitude exactly rather than by an ad hoc numerical rescaling.

The finite horizon provides a second structural simplification. Since the decision graph is acyclic, the one-step horizon can be fitted without bootstrapping; after it is frozen, the two-step horizon has a stationary target, and so forth. Full-action targets exploit the fact that both feasible successors are analytically available. The action-gap term then concentrates additional approximation effort on the quantity that actually determines the greedy decision. These choices make the method closer to approximate dynamic programming specialized to an exact model than to black-box online deep RL.

7.3. What the Numerical Results Do and Do Not Show

The principal conclusion is a reduction in suboptimality, not exact recovery. On all three test families, the seed-averaged Deep-FQI gap is significantly smaller than both feasible baselines under paired instance-level analysis. The strongest relative reductions occur in the extrapolation family, but this should not be interpreted as evidence that extrapolation is universally easier. Absolute reliability gaps depend strongly on the value of r and on the scale of attainable terminal reliability. For this reason the paper reports absolute gaps, recovery rates, and paired differences rather than only percentage improvements.

The exact-design recovery rates remain modest. In particular, the principal test recovery rate is only 5.67%. This prevents any characterization of the learned policy as an exact or near-certain optimizer. The appropriate claim is that it is a reusable approximate policy that, on the tested distributions, reduces design regret relative to low-cost baselines. Exact Pareto DP remains the method of choice when a certificate is required.

7.4. Role of the Heterogeneous Cost Suffix

The explicit counterexample in Section 4 is important for both theory and implementation. It shows that a scalar budget, horizon, and current price do not form a sufficient shared observation. Future prices can reverse the optimal current action even when all those quantities agree. The present network therefore receives the complete ordered normalized suffix. This is an exact sufficient bounded-horizon representation, but its dimension grows linearly with the remaining horizon.

For substantially longer chains, a sequence model may be preferable. A recurrent encoder, temporal convolution, or attention mechanism could compress the suffix into a fixed-width representation. Such an encoder would no longer be an exact hand-derived sufficient statistic, however, and would have to be validated for information loss. The bounded-horizon full-suffix model used here avoids that ambiguity in the first study.

7.5. Potential Field-Data Sources and External Validation Pathway

The present dataset is *synthetic*; no measurements were collected from operational installations. Nevertheless, the model can guide the construction of a future empirical dataset if a physical installation can be abstracted as a sequence of square modules with the prescribed linear/kink attachment alternatives. Candidate settings include modular fiber-optic routes, industrial sensor-network layouts, and structured cabling in buildings. These are prospective sources of measurements, not sources used in the experiments reported here.

For each candidate installation, a field dataset would require: (i) an audited connectivity diagram identifying the source, terminal, individual links, and their attachment order; (ii) dated inspection, maintenance, alarm, or operational logs sufficient to define link functionality during a common observation interval; (iii) installation records or engineering quotations for the incremental cost κ_i of selecting a kink rather than the corresponding linear attachment; and (iv) an explicitly documented budget B under the same cost convention. Repeated link-level observations could estimate a common operational probability r for the period of interest only where pooling links is empirically justified. Such measurements would additionally need checks of missing data, unequal observation exposure, common-cause failures, and dependence between links. A fitted model with link-dependent probabilities or correlated failures would generally require a new reliability derivation; it cannot be treated as a direct validation of the current homogeneous independent-edge equations.

Before comparing an empirical layout with the model, its graph must be checked for compatibility with the polyomino-chain topology and the two permitted attachment decisions. The graph's "directed" label here concerns its growth orientation, whereas terminal connectivity is evaluated on the underlying undirected operational network. If an installation is not representable in this manner, it cannot supply an exact real-world instance of the present model without further modeling assumptions. A meaningful external evaluation would then estimate model parameters on a separate calibration subset, construct feasible candidate designs using observed cost differences, and compare predicted and observed terminal connectivity on held-out periods or installations. No such field evaluation is claimed in this study.

https://doi.org/10.3390/a19121051

7.6. Limitations

Several limitations delimit the conclusions. In particular, no longer-horizon learned-policy evaluation was conducted, and the fixed $h + 5$ observation width grows with the remaining cost suffix. Training separate horizon modules up to $h = 8$ does not establish stability, computational advantage, or policy accuracy for $h > 8$. The observed low exact-recovery rates further preclude claims of reliable exact-design recovery or uniform action correctness. All final learned-policy tests use $m = 8$. The exact solver and the theoretical recursions apply more broadly, but learned heterogeneous generalization at longer horizons remains an open empirical question. Second, the training and test cost distributions, although heterogeneous, are generated from specified smooth or bimodal families. No claim is made for arbitrary industrial cost processes. Third, the current Pareto implementation uses straightforward pairwise dominance checks; computational-geometry data structures may change its practical scaling substantially. Fourth, the statistical tests establish differences on the sampled test families, not universal dominance of Deep FQI over myopic planning.

Finally, the operational-edge probability is common across uncertain links. Allowing edge-specific reliabilities or uncertainty in r would enlarge the physical model and may alter the low-dimensional state structure. Such an extension should be derived from the underlying transfer representation rather than inserted only at the learning layer.

8. Conclusions

We developed a unified exact-and-learning framework for cost-constrained two-terminal reliability design of directed polyomino-chain networks. The exact contribution is based on the linear three-dimensional reliability dynamics: continuation words become alpha-vectors, componentwise dominance gives safe pruning, and heterogeneous costs lead to a joint cost–alpha Pareto recursion. A single exact frontier can answer both hard-budget and nonnegative soft-penalty queries.

The same structure yields positive homogeneity of the value function and scale invariance of the optimal policy. For heterogeneous costs, the ordered remaining cost suffix is genuinely state-relevant; remaining budget and current cost alone are insufficient for a shared policy. These facts motivated a normalized, horizon-wise Deep FQI method using exact model transitions but no optimal-action labels.

Across five independent training seeds and 160 unseen heterogeneous instances, the learned policy consistently reduced mean absolute optimality gap relative to myopic and random feasible baselines, with paired instance-level tests and bootstrap intervals supporting the improvement. Exact recovery remained modest, so the learned policy is best viewed as an amortized approximate decision rule, complementary to rather than a replacement for exact Pareto dynamic programming.

Future work should address longer heterogeneous horizons, compressed representations of remaining cost sequences, faster Pareto-frontier maintenance, and hybrid schemes in which a learned policy proposes high-quality actions while exact cost–alpha optimization is used for selective refinement or certification.

Author Contributions: Conceptualization, M.K.A., A.H.A., L.A. and N.J.R.; methodology, M.K.A., A.H.A., L.A. and N.J.R.; software, M.K.A., A.H.A., L.A. and N.J.R.; validation, M.K.A., A.H.A., L.A. and N.J.R.; writing original draft preparation, M.K.A., A.H.A., L.A. and N.J.R.; writing review and editing, M.K.A., A.H.A., L.A. and N.J.R. All authors contributed equally to this work. All authors have read and agreed to the published version of the manuscript

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Ball, M.O. Computational complexity of network reliability analysis: An overview. *IEEE Trans. Reliab.* **1986**, *35*, 230–239. [CrossRef]
2. Colbourn, C.J. *The Combinatorics of Network Reliability*; Oxford University Press: New York, NY, USA, 1987.
3. Azami, L.; Jafari-Rad, N. Gutman index of polyomino chains. *J. Discret. Math. Appl.* **2025**, *10*, 375–392. [CrossRef]
4. Sigarreta, S.C.; Sigarreta, S.M.; Cruz-Suárez, H. On degree-based topological indices of random polyomino chains. *Math. Biosci. Eng.* **2022**, *19*, 8760–8773. [CrossRef] [PubMed]
5. Yarahmadi, Z.; Ashrafi, A.R.; Moradi, S. Extremal polyomino chains with respect to Zagreb indices. *Appl. Math. Lett.* **2012**, *25*, 166–171. [CrossRef]
6. Azami, L.; Jafari Rad, N. Two-terminal reliability of Markov-dependent directed polyomino-chain networks and its relation to the Gutman index. *J. Stat. Model. Theory Appl.* **2026**. [CrossRef]
7. Azami, L.; Jafari Rad, N. Extremal two-terminal reliability of directed polyomino-chain networks. *J. Stat. Model. Theory Appl.* **2026**, *7*, 219–235. .
8. Zhang, Q.; Liu, Y.; Xiang, Y.; Xiahou, T. Reinforcement learning in reliability and maintenance optimization: A tutorial. *Reliab. Eng. Syst. Saf.* **2024**, *251*, 110401. [CrossRef]
9. Feng, Q.; Wu, Q.; Hai, X.; Ren, Y.; Wen, C.; Wang, Z. Deep reinforcement learning-based resilience optimization for infrastructure networks restoration with multiple crews. *Front. Eng. Manag.* **2025**, *12*, 141–153. [CrossRef]
10. Shahab, E.; Taleb, M.; Gholian-Jouybari, F.; Hajiaghaei-Keshteli, M. Designing a resilient cloud network fulfilled by reinforcement learning. *Expert Syst. Appl.* **2024**, *255*, 124606. [CrossRef]
11. Ernst, D.; Geurts, P.; Wehenkel, L. Tree-based batch mode reinforcement learning. *J. Mach. Learn. Res.* **2005**, *6*, 503–556.
12. Riedmiller, M. Neural fitted Q iteration—First experiences with a data efficient neural reinforcement learning method. In *Machine Learning: ECML 2005*; Lecture Notes in Computer Science 3720; Springer: Berlin/Heidelberg, Germany, 2005; pp. 317–328.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.